Poster: I/O Workload Analysis with Server-side Data Collection

Andrew Uselton and Daniela Ushizima
Lawrence Berkeley National Laboratory
1 Cyclotron, Berkeley, CA
+1 510 486-4061
{acuselton,dushizima}@lbl.gov

ABSTRACT
A detailed understanding of high-performance computer (HPC) file system read and write (I/O) workloads allows stakeholders to evaluate the effectiveness of the I/O infrastructure and identify bottlenecks and other issues. Always-on, server-side monitoring, like that provided by the Lustre Monitoring Tool[1], permits a comprehensive and nonintrusive mechanism for capturing details of the I/O workload. The statistical properties of data movement to and from mass storage on an HPC system reveal transaction patterns that connect the server-side observations back to the computer-side jobs that caused them. This paper lays out strategies to characterize such patterns using I/O statistics.

Categories and Subject Descriptors
D.3.2 [**High performance I/O**]: Analysis of Lustre Monitoring tool

General Terms
Management, Measurement.

Keywords
Feature extraction, data analysis and visualization.

1. INTRODUCTION
Computations are often limited by the growing gap between processor and memory speed [1], particularly storage resources. In order to improve the system performance, it is important to conduct investigation of the read-write to the file system (I/O) and to use policies for workload balance. Monitoring programs for high performance computer (HPC) resources can generate massive amounts of data; processing of this data often requires real-time computations for quick detection of anomalous behavior of the system. Monitoring HPC resources, particularly data storage, is a critical issue to avoid contentions – these undermine the performance of the computing systems because I/O requests cannot be handled fast enough. I/O contentions mean CPU hours lost to the users, and decreased performance of the system.

An example of HPC monitoring scheme is the Lustre Monitoring Tool [2, 3], which records server-side data rates for the storage resources of the HPC system called Franklin. This system employs the Lustre parallel file system as its temporary file resource, using Object Storage Servers (OSS) on which Object Storage Targets (OST) mediate access to large RAID arrays also known as "logical units". Franklin is a Cray XT4 with 38,128 compute cores, 78 TB of memory, 436 TB of disk, and the Cray "Seastar" high-speed internal network. Franklin, an earlier-generation Cray computer, augments the Hopper system. It belongs to the National Energy Research Scientific Computing Center (NERSC), an open-science, high performance scientific computing center sponsored by the U.S. Department of Energy, with approximately 4,000 users.

NERSC deployed the Lustre Monitoring Tool (LMT) for collecting server-side I/O statistics, aimed at identifying I/O contention, quantify its variability and minimize its impact on HPC systems. I/O volume transfers could be monitored from the server or client-side, however client-side statistics are memory intensive, therefore prohibitive at large scale. The LMT has allowed gathering I/O workload on Franklin Lustre-based scratch file system for the past three years.

This paper proposes an algorithm that uses LMT data streams to detect I/O transactions to be correlated to the job-log database, and visualization schemes to tackle large spatio-temporal data from monitoring programs. Our goal is to use these tools in the detection of anomalous data transfers as well as the identification of I/O contention due to load unbalancing. We explain the algorithms to extract I/O transactions from the LMT time series in Section 2 and illustrate our approach with results in Section 3. Conclusions and discussion about the impact of our results and future developments are in Section 4.

2. MAPPING SIGNALS INTO EVENTS
This research exploits server-side observations – how much data was moved in or out in a given 5 second window. The statistics of those observations demonstrate a pervasive under-performance of a large fraction of the system workload. This analysis documents both the breadth and the severity of the problem on one HPC system using the Lustre parallel file system. In addition, we present a new visualization scheme for the time series of I/O observations. This scheme organizes and displays intricate numerical content in a meaningful and easy to understand graphical representation.

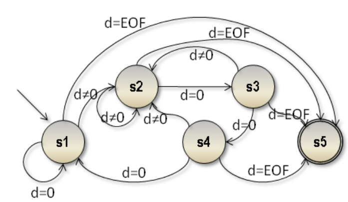

Figure 1.State-diagram of event detection.

The long term average data rate to or from a Lustre server (OST) is about 10MB/s. Only about 5% of all observations are above this average, that is, the observations are “bursty.” An “event” is a contiguous sequence of observations that contain points above the long term average. Figure 1 illustrates the event detection concept, where the time series can be segmented into a sequence of events, with each characterized by the tuple (start time, duration, size). The total amount of data moved in such events accounts for the majority of all I/O to or from the file system.

3. STATISTICS OF EVENTS

After calculating histograms of events organized by length, size, and rate, we noticed that the vast majority of events are transient, occupying only a single observation. The average length of an event is 2 observations (ten seconds), but half of all I/O takes place in events lasting more than four observations. For the smallest events there is an exponential decline in prevalence with increased size. The histograms show that there are several peaks in the distribution for larger values of length, size or rate. Each peak represents a preferred mode of I/O, for example size 2GB is much more likely than slightly larger or smaller sizes. The prevalence of events in the peak indicates a characteristic of a particular application’s I/O pattern or of the workload.

We notice that a large fraction of all I/O is taking place well below the optimum rate of 400MB/s. This fact has implications for the design of applications generating the I/O workload. Poorly organized I/O may use up all the available bandwidth while still achieving suboptimal performance - a competing transaction would not find the extra bandwidth available. To the extent that this sub-optimality is unavoidable it needs to be factored into the design of the I/O subsystem. A job running its I/O at 100MB/s represents four times the opportunity cost compared to running at the maximum available I/O rate.

4. PRELIMINARY RESULTS

Our preliminary findings aimed at mapping the signals into features that can be used to group events. Also, we designed graphical representations as those in Figures 2, which present 48 time series for the Franklin /scratch read (respectively write) values as spark-lines. The x-axis represents 24 hours, and the y-axis gives one spark line for each OST. Each segment indicates a transaction’s duration, with its width proportional to the amount of I/O. This visualization clearly reflects the coordinated timing of some activity across the entire I/O subsystem. This is a fast and easy way to identify correlations between simultaneous events across OSTs. The framework uses the R Statistical tools, particularly the package *multicore* for parallel calculation of transactions given multiple OSTs, and package *fields* for appropriate color to be associated with different OSTs.

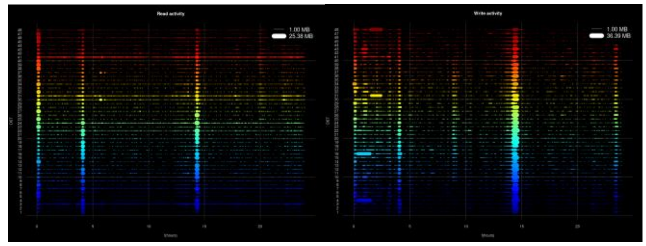

Figure 2. Correlation among OSTs: read and write activity on Franklin using sparklines to indicate I/O transfer volume.

The present analysis identifies events by their timing, duration, and the amount of I/O involved [4]. Those characteristics can often emphasize the I/O of a particular application. This scheme allows calculation of metrics on a continuous basis, allowing dynamic recognition of changes in the workload. The identification still requires a human to search for patterns. Future releases will address algorithms and information visualization regarding the likelihood of a particular user to be associated with transactions which performed large amounts of I/O, e.g. clustering of users based on typical I/O transactions. Another possibility is to identify events that are periodic, as appears to be the case in several data entries.

5. ACKNOWLEDGMENTS

This work was supported by the Director, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

6. REFERENCES

[1] E. Im, I. Bustany, C. Ashcraft, J. Demmel, K. A. Yelick. Performance Tuning of Matrix Triple Products Based on Matrix Structure. In Proceedings of PARA'2004. pp.740-746, 2004.

[2] C. M. Herb Wartens, Jim Garlick. LMT - The Lustre Monitoring Tool. http://code.google.com/p/lmt

[3] A. Uselton, K. Antypas, D. Ushizima, J.Sukharev, "File System Monitoring as a Window Into User I/O Requirements", CUG-2010, Edinburgh, UK, May 24-27th, 2010.

[4] D. Ushizima, A. Uselton, K. Antypas, J. Sukharev "Minimizing I/O contention at NERSC using data analysis", Workshop on Algorithms for Modern Massive Data Sets (MMDS'10), Stanford, CA, June 15-18, 2010